

Examen court de système d'exploitation

Département d'informatique – IUT Villetaneuse

Jeudi 6 novembre 2008 – 1H

Remarques. Seuls les documents suivants sont autorisés :

<http://www.lipn.fr/~cerin/SE/sed.html>, le sujet et la correction du contrôle court de mars 2008, les corrections des TD/TPs. Le barème est indicatif.

1 Langage de commande bash - 10 pts

Écrire un programme Bash qui prend en paramètre trois entiers a, b, r et qui affiche le cercle de centre $C(a, b)$ et de rayon r dont les équations paramétriques sont :

$$\begin{cases} x = a + r \cos \Theta \\ y = b + r \sin \Theta \end{cases}$$

Dans ces équations : $0 \leq \Theta \leq 2\pi$ (l'angle est exprimé en radians). On obtiendra par exemple le graphique suivant pour l'appel `cercle 5 5 5` :

(0,0) est le coin supérieur gauche

```
|-----  
|   XXXXX  
|  XX   XX  
| X     X  
|XX     XX  
|X      X  
|X     .  X  
|X      X  
|XX     XX  
| X     X  
|  XX   XX  
|   XXXXX  
|  
|port-cerin:$
```

Vous utiliserez ou vous vous inspirerez des commandes suivantes :

`tput cols`

Retourne le nombre de colonnes de la fenêtre courante

`tput lines`

Retourne le nombre de lignes de la fenêtre courante

`tput cup m n`

Envoie la commande écran qui place le curseur en ligne m colonne n . Par exemple, `tput cup 0 0` place le curseur en position (0,0) (le coin en haut à gauche de l'écran, position de curseur appelée 'home')

`tput clear`

Envoie la commande d'effacement de l'écran au terminal courant.

`x='echo "3 + 4 * c(2)" | bc -l'`

Range dans x la valeur $3 + 4 \cdot \cos(2)$, L'angle est exprimé en radians. Un angle de 1 rad est un angle, qui, ayant son sommet au centre d'un cercle, intercepte, sur la circonférence de ce cercle, un arc d'une longueur égale à celle du rayon du cercle. Un cercle complet représente un angle de $2 \cdot \pi$ radians, appelé angle plein.

`y='echo "3 + 4 * s(2)" | bc -l'`

Range dans y la valeur $3 + 4 \cdot \sin(2)$, L'angle est exprimé en radians.

`v='python -c "import sys;print round(float(sys.argv[1]))" "12.34"'`

Assigne à v la valeur arrondie de 12.34 (ici, $v=12$).

`v='python -c "import sys;print round(float(sys.argv[1]))" "12.54"'`

Assigne à v la valeur arrondie de 12.54 (ici, $v=13$).

2 Langage de commande bash - 10pts

a) On vous demande d'écrire un script bash `numeroter.sh` qui va numéroter les lignes d'un fichier. Attention : la numérotation doit se faire « sur la droite » en étant alignée sur la ligne de plus grande longueur.

Pour le fichier `ListeGroupes.csv` le programme affichera :

```
$ more ListeGroupes.csv
10501311,Mr,B0,DELNEUF,MAXIME
10602789,Mr,B1,DELPIERRE,JEREMY
10601320,Mr,A0,DESSAINT,FLORENT
10601814,Mr,A1,DUARTE,MATHIEU
10603349,Mr,A1,DUFAG,ALEXANDRE
10601894,Mr,A0,DUMONTIER,STEPHANE
10601882,Mr,A2,DURAND,PIERRE
10601945,Melle,B0,BELALIMAT,LEILA
$ /bin/bash numeroter.sh ListeGroupes.csv
-----Fichier en entree: ListeGroupes.csv-----
10501311,Mr,B0,DELNEUF,MAXIME      1
10602789,Mr,B1,DELPIERRE,JEREMY   2
10601320,Mr,A0,DESSAINT,FLORENT   3
10601814,Mr,A1,DUARTE,MATHIEU     4
10603349,Mr,A1,DUFAG,ALEXANDRE    5
10601894,Mr,A0,DUMONTIER,STEPHANE 6
10601882,Mr,A2,DURAND,PIERRE      7
10601945,Melle,B0,BELALIMAT,LEILA 8
```

b) Expliquez pourquoi cela est plus compliqué que de numéroter à gauche, puis proposez une solution avec Sed pour numéroter les lignes sur la gauche.

3 Proposition de correction de la question 1

```
#!/bin/bash
#
#echo $1
#echo $2
#echo $3

clear
if test $# = 3
then
    # calcul de pi
    pi='echo 'scale=10;4*a(1)' | bc -l'
    angle=0.0
    increment='echo "2*${pi} / 100 "|bc -l'
    while [ 'echo "$angle > 2*${pi}"| bc -l' -lt 1 ];
    do
        x='echo "$1 + $3 * c($angle)"|bc -l'
        x='python -c "import sys;print int(round(float(sys.argv[1])))" $x'
        y='echo "$2 + $3 * s($angle)"|bc -l'
        y='python -c "import sys;print int(round(float(sys.argv[1])))" $y'
        tput cup $x $y
        echo -n "X"
    #         echo $x $y $angle
        angle='echo "$angle + $increment"|bc -l'
    done
    # on dessine le centre du cercle
    tput cup $1 $2
    echo -n "."
    # on se déplace en bas de la fenetre
    x='tput lines'
    tput cup $((x-2)) 'tput cols'
    echo
else
    echo "Usage - $0 x y r"
    echo "    x: abscisse, y ordonnee, r: rayon"
    echo "    du cercle"
fi
```

Il reste à écrire les conditions pour ne pas écrire en dehors de la fenêtre.

4 Proposition de correction de la question 2

```
#!/bin/bash

if [ ! -e $1 ]; then
    echo "le fichier n existe pas!"
    exit 0
fi
echo "-----Fichier en entree: $1-----"
max=0
cat < $1 | while true    #pour toutes les lignes du fichier
do
    read ligne
    if [ "$ligne" = "" ]; then break; fi
    # traitement de $ligne
    echo $ligne
    myLong=${#ligne}
    if [ $((myLong)) -gt $((max)) ]
    then
        max=$myLong
        echo $max > /tmp/max
    fi
done

# La longueur de la ligne de plus grande longueur est
# contenue dans le fichier /tmp/max

i=1
max='less /tmp/max'
cat < $1 | while true    #pour toutes les lignes du fichier
do
    read ligne
    if [ "$ligne" = "" ]; then break; fi
    # traitement de $ligne
    #echo $ligne $max
    myLong=${#ligne}
    diff='expr $((max)) - $((myLong))'
    echo -n $ligne
    # on affiche maintenant les blancs
    for (( j=0 ; j<$((diff)) ; j++ ))
    do
        echo -n " "
```

```

done
# on affiche le numero de ligne
echo " $i"
i=$((i+1))
done

```

```
rm /tmp/max
```

Nous devons effectuer deux passage sur le fichier : le premier pour calculer la ligne de plus grande longueur puis pour effectuer la numérotation.

Pour numéroter sur la gauche avec Sed, on peut lancer la commande :

```

# numéroter chaque ligne du fichier (appuyé simplement à gauche).
# L'utilisation d'une tabulation (voir la note sur '\t' à la fin
# de ce texte) au lieu d'un espace préservera les marges.
sed = nomdefichier | sed 'N;s/\n/\t/'

```

```

# numéroter chaque ligne d'un fichier (numéro à gauche, appuyé à droite)
sed = nomdefichier | sed 'N; s/^/ /; s/ *\(.{6,}\)\n/\1 /'

```

```

# numéroter chaque ligne d'un fichier, mais afficher le numéro de ligne
# seulement si la ligne n'est pas vide.
sed '/./=' nomdefichier | sed '/./N; s/\n/ /'

```

Nous proposons maintenant une autre solution au problème à partir d'une définition récursive : on calcule récursivement la longueur de la ligne qui comprend le plus de caractères, puis on numérote sur la droite à partir de la valeur du deuxième paramètre de la fonction `numbered_line`. On notera au passage l'utilisation de `tac` - Concaténer et afficher des fichiers à l'envers. Il reste un léger problème avec ce code : les caractères de tabulation ou les blancs en début de ligne sont ignorés. veuillez remédier à la situation.

```
#!/bin/bash
```

```

if [ ! -e $1 ]; then
    echo "le fichier n existe pas!"
    exit 0
fi

```

```

# numbered_line INIT_MAX INIT_COUNT < FILE
numbered_line() {
    local line=""

```

```

local i=0
local max=$1
local count=$2

# traitement de chaque ligne
if read line
then
    local size='echo $line | wc -c'
    if [ $max -lt $size ]
    then
        numbered_line $size [$count+1]
        max=$?
    else
        numbered_line $max [$count+1]
        max=$?
    fi
    # la plus grande longueur est $max
    echo -n $line
    # affichage des espaces manquants
    for i in `seq $[ $max - ${#line} ]`
    do
        echo -n " "
    done
    # Affichage Compteur
    echo $count
    return $max
else
    return $max
fi
}

echo "-----Fichier en entree: $1-----"
numbered_line 0 1 < $1 | tac

```